Parsons Problem Website

Group 004
Chenwei Gu, Peihong Yao, Xuanyu Liu, Yichen Zhang, Yuhao Liu

Summary

The Pandas Learning Website is designed to help students learn pandas by providing interactive learning tools. Students have the ability to choose topics and tags, which are then sent to a provided AI API for the generation of Python code examples. These examples are subsequently transformed into Parsons Problems. These problems help students understand and practice pandas by rearranging code snippets into the correct order. The site will also allow administrators to manage content, and regular users are not required to login.

Specifications

Features

1. Topic and Tag Selection
    - **User Interface**: Drop-down menus or multi-select boxes for users to choose pandas topics (e.g., dataframe, data visualization, classification) and relevant tags (e.g., sports, games, food).
    - **Implementation**: Frontend forms capture user selections and send data to the backend.
2. ChatGPT Integration
    - **User Interface**: A button to generate code examples based on selected topics and tags.
    - **Implementation**: Use provided AI API to generate Python code. The API call uses selected topics and tags to generate context-appropriate code. In the early stages, can manually fetch code from OpenAI. However, it should have ability to integrate OpenAI as soon as needed.
3. Parsons Problem Generation
    - **User Interface**: Draggable code snippets for users to arrange in the correct order.
    - **Implementation**: Backend generate code and sends it to the frontend. Frontend use some existing libraries (e.g. js-parsons) shuffle up code into Parsons Problems. Future implementation includes adding misleading options for higher difficulty levels.
4. Code Execution and Result Display
    - **User Interface**: Submit button for users to submit their arranged code and display area for execution results.
    - **Implementation**: Backend runs the submitted code returns the results to the frontend.
5. Administration Account

- **User Interface**: Admin login page and dashboard for managing topics/tags.

Technology Stack

1. **Frontend**: JavaScript, HTML, CSS
2. **Backend**: Python/Java

User Story

As a	I want to	So that	Prioritization	Func / Non-func
Student	Select the topic of questions before self-practice	I can focus working on the topic which I need or want to review	High	Functional
Student	access the website without needing to create an account	I can start learning immediately.	High	Functional
Student	generated Python code example	I can study how pandas functions are applied in specific contexts.	High	Functional
Student	engage with Parsons Problems by arranging shuffled code snippets in the correct order	I can deepen my understanding of panda	High	Functional
Student	see the IDE result of my arranged code execution	I can verify if my understanding is correct.	Mid	Functional
Student	check the answer If I can't solve the problem,	I can know the right answer	High	Functional
Student	see the result (Yes/No) of my arranged code execution	I can verify if my understanding is correct.	High	Functional
Student	Select the difficulty level of questions before practicing	From the simple into the difficult questions can let me better review and study	Low	Functional
Student	Check the accuracy of this exercise	I can know if I can making progress	Low	Functional
Student	Have a timer	know how much time I spend on the problem	Mid	Functional
Student	Cookey store analytics	I can know my progress	Low	Functional
Teacher	share a specific problem with my students	they can work on it and improve their understanding of a particular pandas concept.	Low	Functional

Owner	Make the user interface to be clearly and concisely	Users can easily to use the website for practice or sharing	High	Non-Functional
Administrator	log in to a dashboard to manage topics, tags, and difficulty levels	Website can keep content relevant and challenging.	Low	Functional
Owner	Supports concurrent access by 1000 users	Websites can be used by multiple people at the same time	Mid	Non-Functional

Project timelines

Sprint	*Week*
1: Project Planning & Setup	1-2
2: Requirement Doc & Prototype/Demo	3-4
3: Design (UI, Code structure …)	5
4: Coding	6-8
5: Testing	9-10
6: Online	11